

ThreatQuotient

A Securonix Company



VMware Carbon Black EDR Connector

Version 2.2.2

November 24, 2025

ThreatQuotient

20130 Lakeview Center Plaza Suite 400
Ashburn, VA 20147

 **ThreatQ Supported**

Support

Email: tq-support@securonix.com

Web: <https://ts.securonix.com>

Phone: 703.574.9893

Contents

| | |
|-----------------------------------|----|
| Warning and Disclaimer | 3 |
| Support | 4 |
| Integration Details..... | 5 |
| Introduction | 6 |
| Prerequisites | 7 |
| Integration Dependencies | 7 |
| Installation..... | 8 |
| Configuration | 9 |
| ThreatQ Authentication..... | 12 |
| Generating OAuth Credentials..... | 12 |
| Options..... | 14 |
| Starting the EDR Connector | 15 |
| Upgrading the EDR Connector | 16 |
| Debugging the EDR Connector | 17 |
| Change Log | 18 |

Warning and Disclaimer

ThreatQuotient, Inc. provides this document “as is”, without representation or warranty of any kind, express or implied, including without limitation any warranty concerning the accuracy, adequacy, or completeness of such information contained herein. ThreatQuotient, Inc. does not assume responsibility for the use or inability to use the software product as a result of providing this information.

Copyright © 2025 ThreatQuotient, Inc.

All rights reserved. This document and the software product it describes are licensed for use under a software license agreement. Reproduction or printing of this document is permitted in accordance with the license agreement.

Support

This integration is designated as **ThreatQ Supported**.

Support Email: tq-support@securonix.com

Support Web: <https://ts.securonix.com>

Support Phone: 703.574.9893

Integrations/apps/add-ons designated as **ThreatQ Supported** are fully supported by ThreatQuotient's Customer Support team.

ThreatQuotient strives to ensure all ThreatQ Supported integrations will work with the current version of ThreatQuotient software at the time of initial publishing. This applies for both Hosted instance and Non-Hosted instance customers.



ThreatQuotient does not provide support or maintenance for integrations, apps, or add-ons published by any party other than ThreatQuotient, including third-party developers.

Integration Details

ThreatQuotient provides the following details for this integration:

Current Integration Version 2.2.2

Compatible with ThreatQ Versions $\geq 4.30.0$

Compatible with VMware Carbon Black EDR Servers $\geq 6.5.2$

Operating Systems RedHat 6.0, CentOS 7.0

Python Version 3.6

Third-Party Hosting Type On-Prem

Support Tier ThreatQ Supported

Introduction

The VMware Carbon Black EDR connector integration pulls Active indicators from a ThreatQ data collection and imports them into Carbon Black EDR as Threat Reports.



This integration will be installed and run on the Carbon Black EDR server.

The integration configuration allows you customize what information is exported from ThreatQ into Carbon Black EDR by enabling you to specify multiple saved search names.

This Connector will receive data only from saved searches that contain **Indicators** of type:


- IP Address
- IPv6 Address
- MD5
- FQDN
- SHA-256

Prerequisites


The following is required in order to install and run the connector:

- Access Token to connect to your VMware Carbon Black EDR instance.
- Credentials to connect to your ThreatQ instance. See the [ThreatQ Authentication](#) chapter for more details.

Integration Dependencies

 The integration must be installed in a python 3.6 environment.

The following is a list of required dependencies for the integration. These dependencies are downloaded and installed during the installation process. If you are an Air Gapped Data Sync (AGDS) user, or run an instance that cannot connect to network services outside of your infrastructure, you will need to download and install these dependencies separately as the integration will not be able to download them during the install process.

 Items listed in bold are pinned to a specific version. In these cases, you should download the version specified to ensure proper function of the integration.

| DEPENDENCY | VERSION | NOTES |
|--------------------|---------------|---------------|
| cbapi | 1.7.10 | Pinned |
| requests | >=2.27.1 | N/A |
| flask | 1.1.4 | Pinned |
| pyinstaller | 4.10.0 | Pinned |
| threatqsdk | >=1.8.11 | N/A |
| simplejson | 3.19.1 | Pinned |

Installation

The VMware Carbon Black EDR Response Connector is packaged into an RPM file which will be installed on your Carbon Black EDR Response server. You can download the RPM file from the ThreatQ Marketplace.

1. Transfer the RPM file to your Carbon Black EDR Response instance:

```
scp python-cb-threatq-connector-<version>.rpm root@<cb-ip-address>:/tmp/
```

2. SSH into your Carbon Black EDR Response instance.
3. Install the package using the RPM:

```
cd /tmp

rpm -ivh python-cb-threatq-connector-<version>.rpm --ignoreos --nofiledigest
```



The OS is ignored in the command above. If the OS is not ignored, the installation will fail.

You will still need to [configure and then enable the connector](#).

Configuration



ThreatQuotient does not issue API keys for third-party vendors. Contact the specific vendor to obtain API keys and other integration-related credentials.

The connector now needs to be configured on the Carbon Black EDR Response instance side.

1. Create a default "credentials.response" file for the connector to use.

```
[default]
url=https://localhost
token=xxxxxxx
ssl_verify=False
```



Your token can be found in your user profile in the Carbon Black EDR Web UI.

2. Locate the sample of the configuration file at:

```
/etc/cb/integrations/threatq/connector.conf.example
```

3. Copy the sample file to the following pathway:

```
/etc/cb/integrations/threatq/connector.conf
```

4. Configure the ThreatQ connector by entering the following parameters:

```
[auth] #-----
# ThreatQ API configuration
#-----
# This section allows global configuration options to be passed to the ThreatQ feed.

threatq_host=https://<your threatq host>

# Username and password and/or OAuth authentication methods can be configured.
# Username and password having priority if both specified.

# 1
threatq_client_id=
threatq_username=
threatq_password=

# 2
threatq_oauth_client_id=
threatq_oauth_client_secret=

threatq_verify_ssl=false
threatq_http_proxy=
threatq_https_proxy=



[bridge]
#-----
# Core Configuration
#-----
listener_port=6300
listener_address=127.0.0.1
feed_retrieval_minutes=60
```




```
# debug=1

# API key for an administrative user of the Carbon Black EDR server
carbonblack_server_token=
carbonblack_server_sslverify=false

# Only uncomment out the carbonblack_server_url if you are running the connector on a
machine
# *other* than the Cb server itself.
# carbonblack_server_url=

# If you need to use an HTTPS proxy to access the ThreatQ API server, uncomment and
configure the https_proxy # variable below.
# https_proxy=http://proxyuser:proxypass@proxyhostname:proxyport
# http_proxy=http://proxyuser:proxypass@proxyhostname:proxyport
```

| PARAMETER | DESCRIPTION |
|------------------------------------|---|
| threatq_host | Example: https://<> |
| threatq_client_id | Client ID of ThreatQ instance specified above (on threatq_host) <div>  The Client ID can be found on the UI of ThreatQ. </div> |
| threatq_username | Username used to connect to ThreatQ instance. |
| threatq_password | Password used to connect to ThreatQ instance. |
| threatq_oauth_client_id | OAuth client id used to connect to ThreatQ instance. |
| threatq_oauth_client_secret | OAuth client secret used to connect to ThreatQ instance. |
| threatq_saved_search_names | This names can be found on the UI of ThreatQ instance, on the saved search section. <div>  This is a comma-delimited list of saved searches. </div> |

| PARAMETER | DESCRIPTION |
|----------------------------------|---|
| <code>threatq_verify_ssl</code> | <p>This is whether you want to verify the SSL connection between CB Response and ThreatQ. Setting this to true may cause connection issues.</p> <div> This option is set to false by default</div> |
| <code>threatq_http_proxy</code> | <p>This is optional. If you want to communicate with ThreatQ via an HTTP proxy, set it here.</p> <div> This must include the username, password, host/ip, and port.</div> |
| <code>threatq_https_proxy</code> | <p>This is optional. If you want to communicate with ThreatQ via an HTTPS proxy, set it here.</p> <div> This must include the username, password, host/ip, and port.</div> |

ThreatQ Authentication

There are two types of authentication that can be used for connecting to the ThreatQ instance.

First one uses the `threatq_client_id`, `threatq_username`, `threatq_password` for the specified ThreatQ instance.

The second one uses OAuth (`threatq_oauth_client_id`, `threatq_oauth_client_secret`).

OAuth is supported on ThreatQ instance starting with version v4.30.

Notes:

- If fields for both authentication types are completed in the configuration file (section # 1 and # 2 located in the configuration file listed in step 4 of the [Configuration](#) chapter) then priority will have the client_id/user/password authentication, so this one will be used for connecting to the instance
- If for both section from authentication (section # 1 and # 2 located in the configuration file listed in step 4 of the [Configuration](#) chapter) fields are missing or are empty than an error will be raised when the connector service will start
- If just one of the sections from authentication (section # 1 and # 2 located in the configuration file listed in step 4 of the [Configuration](#) chapter) has all fields completed and non empty that this one will be used.

Generating OAuth Credentials

The following details how to generate ThreatQ OAuth Credentials for the integration.



Save these two tokens in a safe and secure location.

ThreatQ v5

1. Navigate to the following directory:

```
cd /var/www/api
```

2. Run the OAuth registration command:

```
sudo php artisan threatq:oauth2-client --name "Carbon Black EDR  
Response Connector" --user_group "analyst"
```

ThreatQ v6

1. Run the OAuth registration command:

```
kubectl exec --namespace threatq --stdin --tty deployment/api-  
schedule-run -- ./artisan threatq:oauth2-client --name="Carbon Black  
EDR Response Connector" --user_group="analyst"
```

Options

The server can be configured to retrieve data from a ThreatQ saved search at specific intervals.

The `[bridge]` section of the configuration file the parameter `feed_retrieval_minutes` must be completed in order to use this option. See [step 4 in the Configuration chapter](#) for more details.

Starting the EDR Connector

Prerequisites

- Connector has been installed via RPM.
- Connector has been configured on the ThreatQ side (created the saved searches).
- Connector has been configured on the Carbon Black EDR Response side (connector.conf).

Run one of the following commands, depending on the CentOS version:

Centos6

```
service cb-threatq-connector start
```

Centos7

```
systemctl start cb-threatq-connector
```

Upgrading the EDR Connector

If the ThreatQ connector has already been installed once and needs to be updated, follow these steps:

1. Transfer the new RPM file to your Carbon Black EDR Response instance:

```
scp python-cb-threatq-connector-2.2.1-10.x86_64.rpm root@<cb-ip-address>:/tmp/
```

2. SSH into your Carbon Black EDR Response instance.
3. Stop the old connector:

```
service cb-threatq-connector stop
```

4. Use the RPM to upgrade your connector:

```
cd /tmp rpm -Uvh --force python-cb-threatq-connector2.2.1-10.x86_64.rpm --ignoreos --nofiledigest
```

5. Start the connector up again:

```
service cb-threatq-connector start
```


Debugging the EDR Connector

If the connector is up and running, but you aren't seeing any indicators within Carbon Black EDR Response, you can debug it by looking at the connector log file:

Log Location: `/var/log/cb/integrations/cb-threatq-connector/cbthreatq-connector.log`

If you want to see the data that is being downloaded by Carbon Black EDR Response, you can directly curl the connector from your Carbon Black EDR Response instance.

1. SSH into your Carbon Black EDR Response instance.
2. Use curl to send a request to the connector

```
curl -k http://127.0.0.1:6300/threatq/json?server_token=<enter your server token here>
```



You can get the server token from the log file specified above. Otherwise, you might just be able to leave it out.

Change Log

- **Version 2.2.2**
 - Updated the integration's required dependencies. threatqsdk version 1.8.11 or greater is now required (was previously 1.8.7).
- **Version 2.2.1**
 - Resolved the following issues:
 - an error would occur if the ThreatQ feed was not present on the Carbon Black EDR instance.
 - required files were missing after upgrading to Python 3.6.
- **Version 2.2.0**
 - Added support for python 3.
 - Added support for SHA-256 indicators.
- **Version 2.0.1**
 - Rebuilt RPM using ThreatQ SDK 1.8.1.
- **Version 2.0.0**
 - Initial Release - Replaced Carbon Black Response Connector